EXAMINATION SYSTEM

Power BI Developer Track

Abdelrahman Mohamed - Asya Fouda - Hadeer Mohamed
Karim Mohsen - Wafaa Hamada

Introduction

Welcome to the documentation for our last project in iti power BI developer tack, which was completed during our 3-month internship at ITI. In this project, we used tools and skills which gained during the internship to build a complete examination system from scratch that provides valuable insights into business performance and helps decision-makers make data-driven decisions.

The goal of this project was to demonstrate our skills and to create dashboards that could be used by a business to monitor their key performance indicators (KPIs). We started by identifying the KPIs that were most important to the business and gathering the data needed to create the dashboard.

Overall, this project allowed us to demonstrate our skills to create a valuable tool for businesses looking to monitor their performance and make data-driven decisions. We hope that this documentation will serve as a useful resource for those interested in BI development and data Analysis.

Methodology

- ## MS-SQL Server

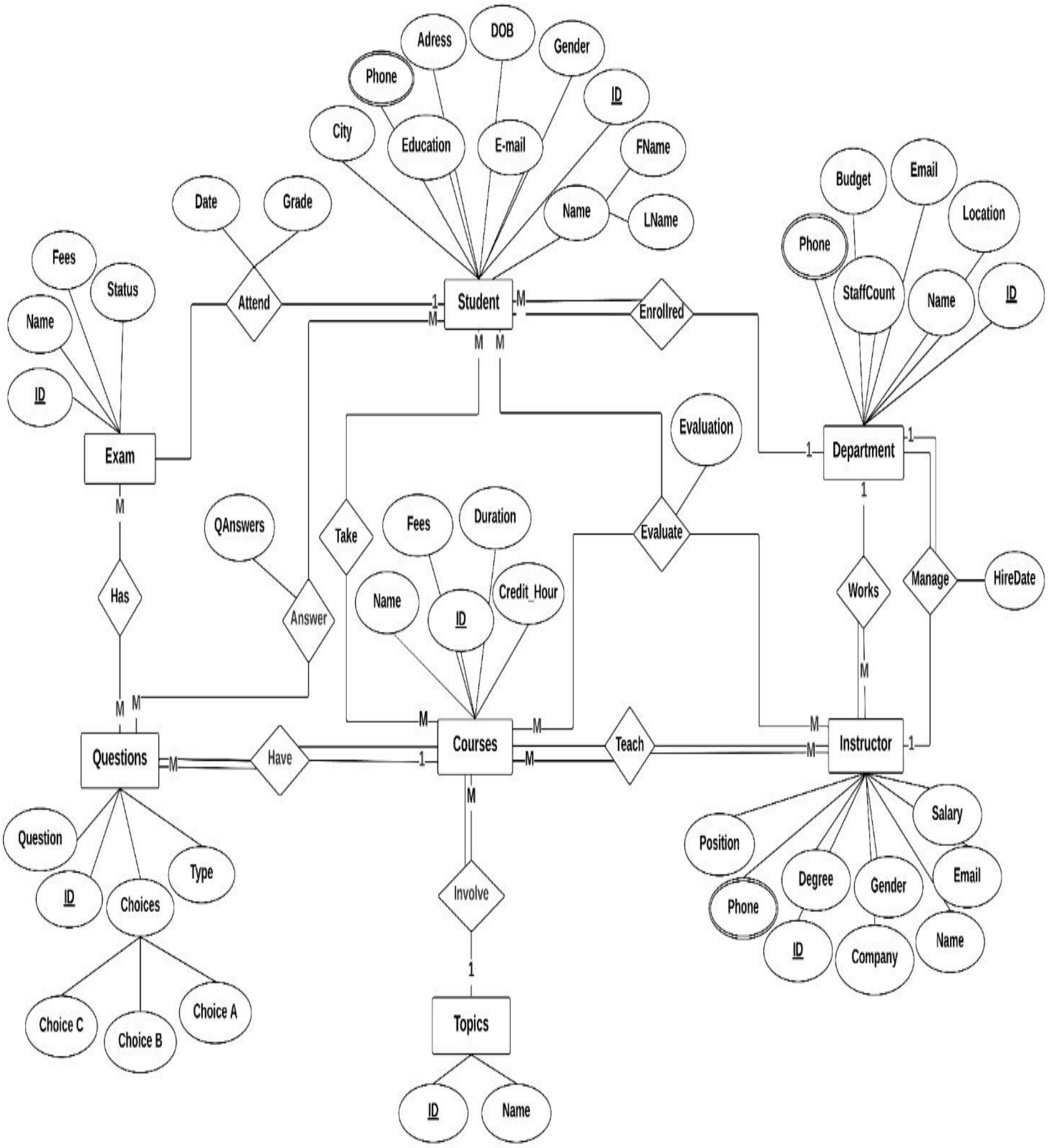
1- Identify the data requirements.

The first step in creating a database for an examination system project is to identify the data requirements. This includes determining what data needs to be stored, how it should be organized, and what relationships exist between different data entities.

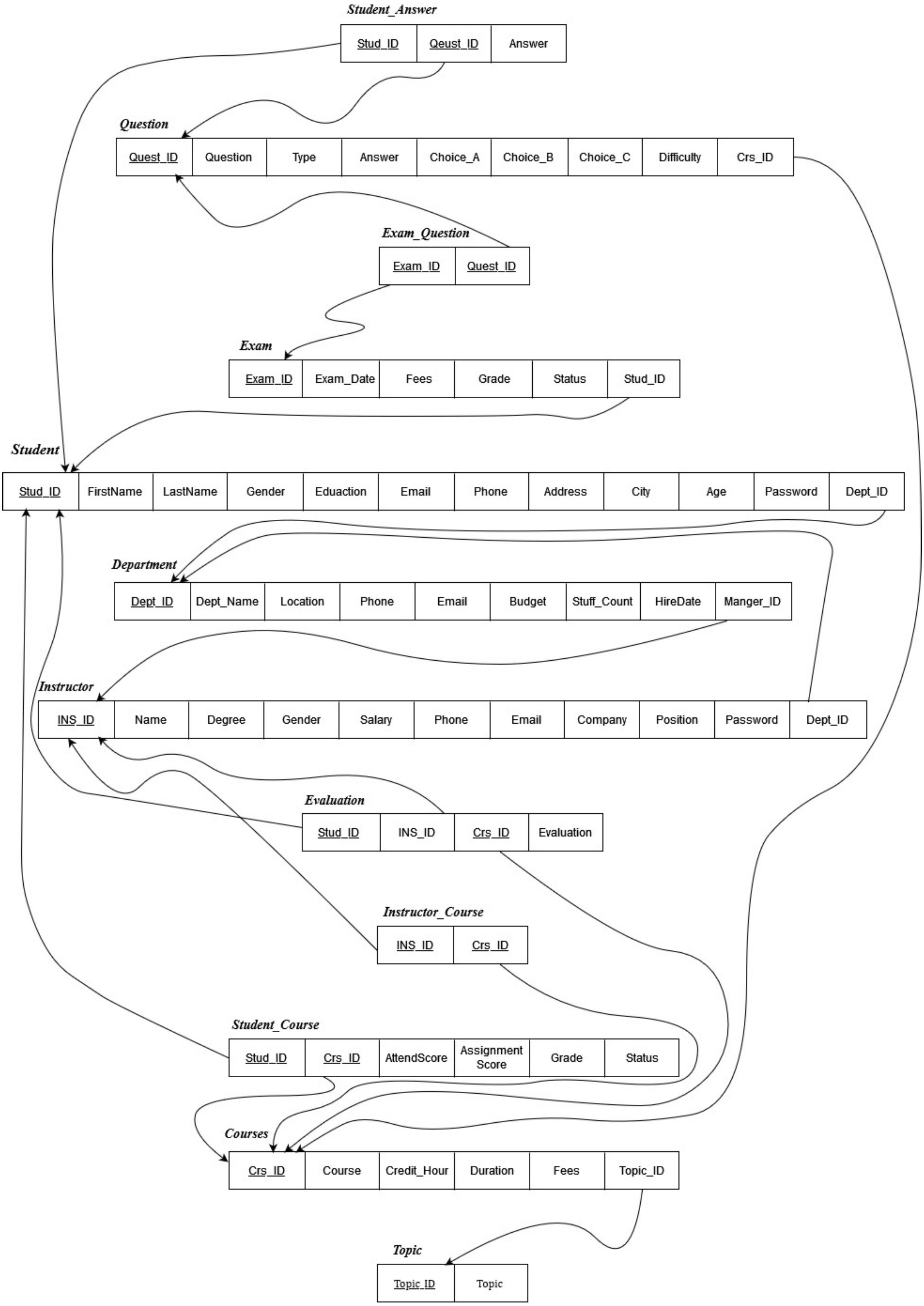
We decided to have 8 entities.

1- Student
2- Department
3- Instructor
4- Courses
5- Topics
6- Exam
7- Questions
8- Evaluation

2- Create an entity-relationship diagram.



3- Design the database schema.

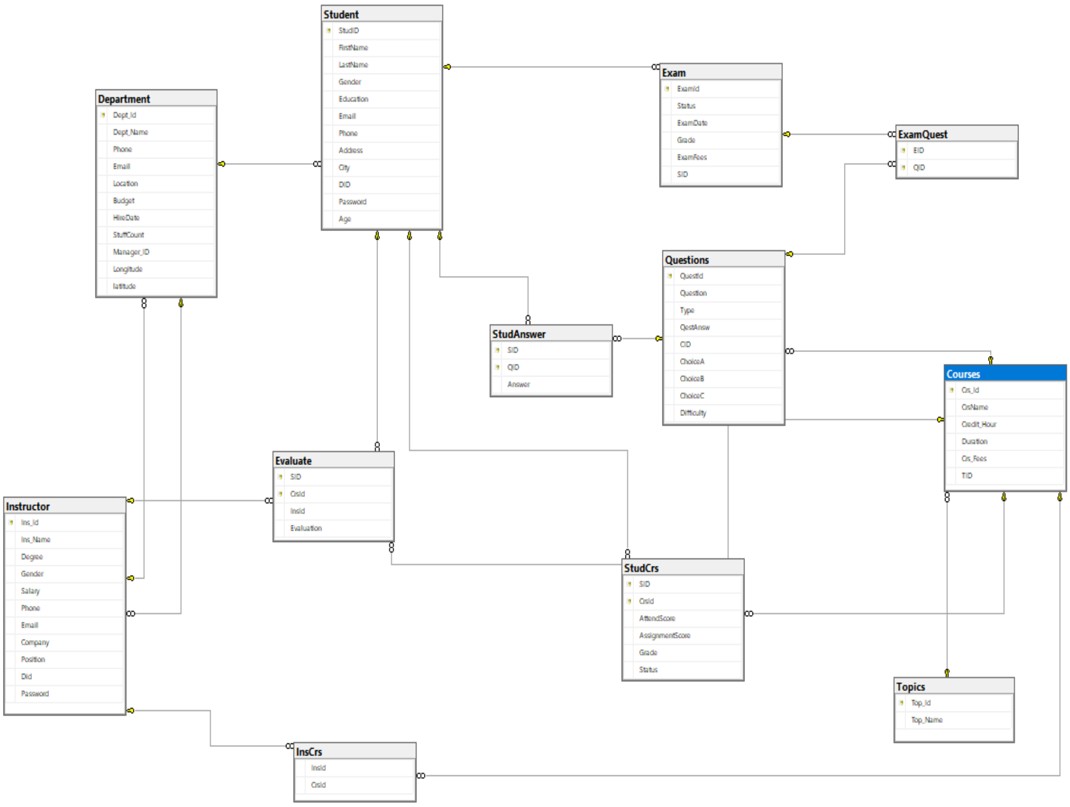


During mapping and for normalization purpose we had 12 tables.

1- Student
2- Department
3- Instructor
4- Courses
5- Topics
6- Exam
7- Questions
8- Evaluation
9- Student Answer
10- Instructor-Course
11- Exam-Question
12- Student-Course

4- Create the database.

The final physical schema for our examination system database



Which contain:

1- Student
    A. Student ID (Primary key)
    B. First Name
    C. Last Name
    D. Address
    E. Age
    F. Email
    G. Phone Number
    H. City
    I. Gender
    J. DID (Foreign key)

2- Department
    A. Department ID (Primary key)
    B. Dept Name
    C. Location
    D. Budget
    E. Email
    F. Phone Number
    G. Manager ID
    H. Hiring Date

3- Instructor
    A. Instructor ID (Primary key)
    B. Instructor Name
    C. Degree
    D. Gender
    E. Email
    F. Phone Number
    G. Position
    H. Salary
    I. Company
    J. Department ID (Foreign key)

4- Courses
    A. Course ID (primary key)
    B. Course Name
    C. Course Fess
    D. Duration
    E. Credit Hours
    F. Topic ID (Foreign key)

5- Topics
    A. Topic ID (Primary key)
    B. Topic Name

6- Exam
   A. Exam ID (Primary key)
   B. Exam Date
   C. Exam Fees
   D. Grade
   E. Status
   F. Student ID (Foreign key)
   G. Course ID (Foreign key)

7- Questions
   A. Question ID (Primary key)
   B. Question
   C. Question Answer
   D. Choice A
   E. Choice B
   F. Choice C
   G. Question Type
   H. Course ID (Foreign key)

8- Evaluation
   A. Student ID (Composite key)
   B. Course ID (Composite key)
   C. Evaluation

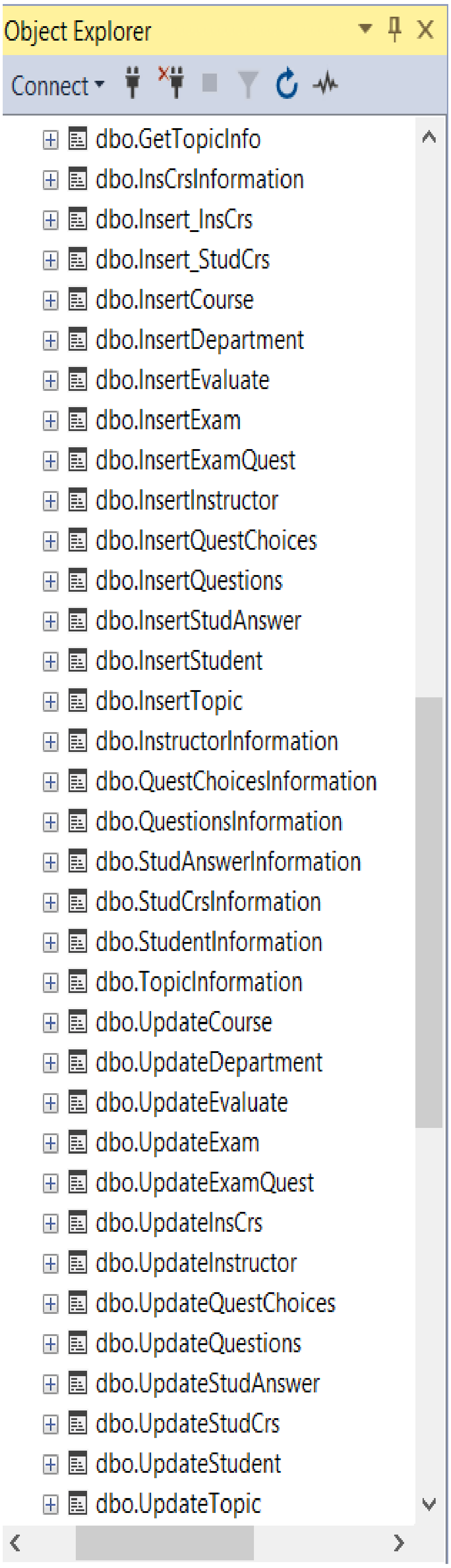
9- Student Answer
   A. Exam ID (Composite key)
   B. Student ID (Composite key)
   C. Answer

10- Instructor-Course
   A. Instructor ID (Composite key)
   B. Course ID (Composite key)

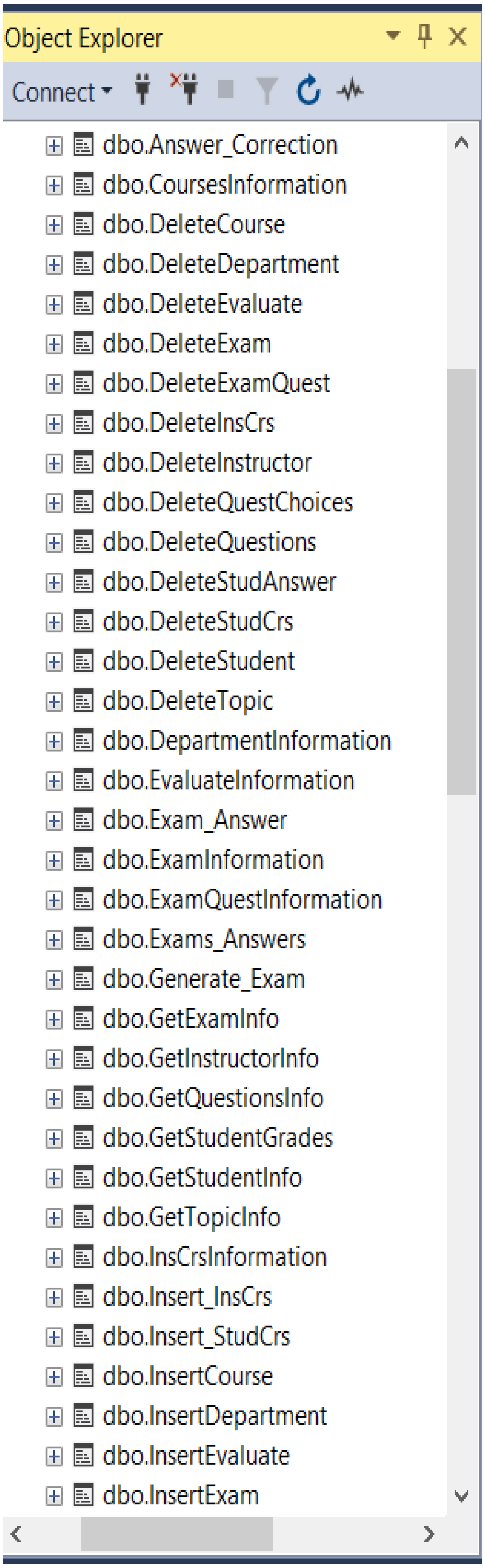
11- Exam-Questions
   A. Exam ID (Composite key)
   B. Question ID (Composite key)

12- Student-Course
   A. Student ID (Composite key)
   B. Course ID (Composite key)

5- Create stored procedures to maintain manipulation of data in database.



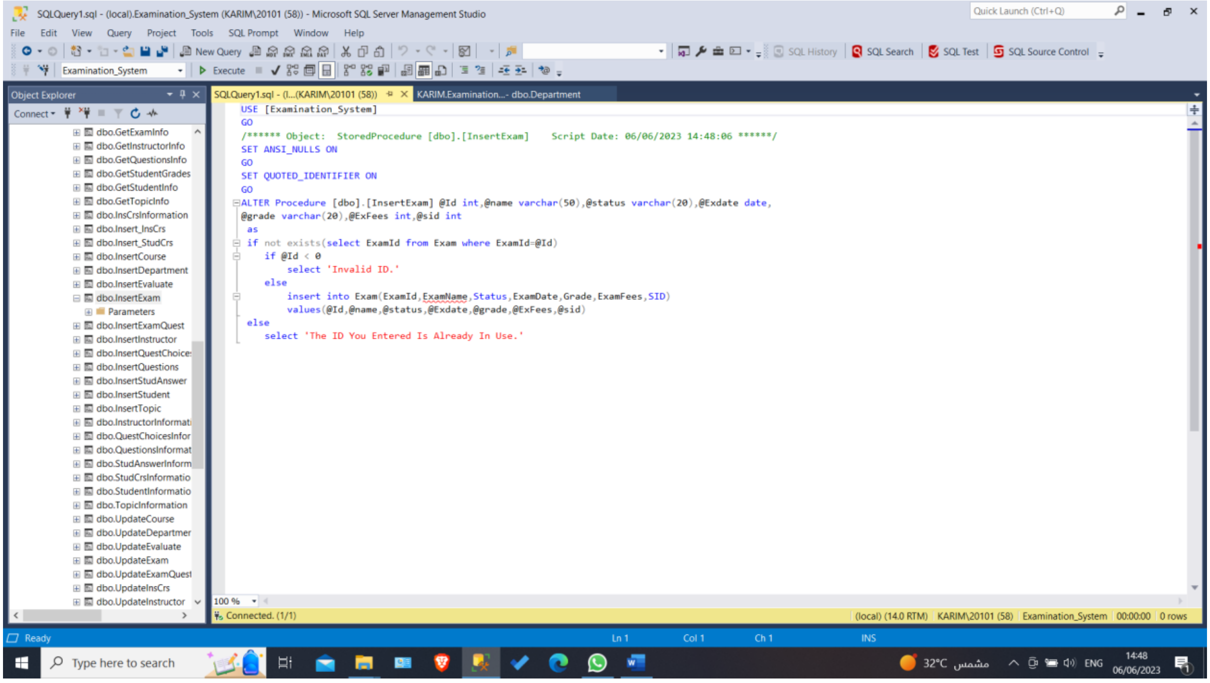
dbo.GetTopicInfo
dbo.InsCrsInformation
dbo.Insert_InsCrs
dbo.Insert_StudCrs
dbo.InsertCourse
dbo.InsertDepartment
dbo.InsertEvaluate
dbo.InsertExam
dbo.InsertExamQuest
dbo.InsertInstructor
dbo.InsertQuestChoices
dbo.InsertQuestions
dbo.InsertStudAnswer
dbo.InsertStudent
dbo.InsertTopic
dbo.InstructorInformation
dbo.QuestChoicesInformation
dbo.QuestionsInformation
dbo.StudAnswerInformation
dbo.StudCrsInformation
dbo.StudentInformation
dbo.TopicInformation
dbo.UpdateCourse
dbo.UpdateDepartment
dbo.UpdateEvaluate
dbo.UpdateExam
dbo.UpdateExamQuest
dbo.UpdateInsCrs
dbo.UpdateInstructor
dbo.UpdateQuestChoices
dbo.UpdateQuestions
dbo.UpdateStudAnswer
dbo.UpdateStudCrs
dbo.UpdateStudent
dbo.UpdateTopic



dbo.Answer_Correction
dbo.CoursesInformation
dbo.DeleteCourse
dbo.DeleteDepartment
dbo.DeleteEvaluate
dbo.DeleteExam
dbo.DeleteExamQuest
dbo.DeleteInsCrs
dbo.DeleteInstructor
dbo.DeleteQuestChoices
dbo.DeleteQuestions
dbo.DeleteStudAnswer
dbo.DeleteStudCrs
dbo.DeleteStudent
dbo.DeleteTopic
dbo.DepartmentInformation
dbo.EvaluateInformation
dbo.Exam_Answer
dbo.ExamInformation
dbo.ExamQuestInformation
dbo.Exams_Answers
dbo.Generate_Exam
dbo.GetExamInfo
dbo.GetInstructorInfo
dbo.GetQuestionsInfo
dbo.GetStudentGrades
dbo.GetStudentInfo
dbo.GetTopicInfo
dbo.InsCrsInformation
dbo.Insert_InsCrs
dbo.Insert_StudCrs
dbo.InsertCourse
dbo.InsertDepartment
dbo.InsertEvaluate
dbo.InsertExam

A. Create 4 Stored procedures for every table.

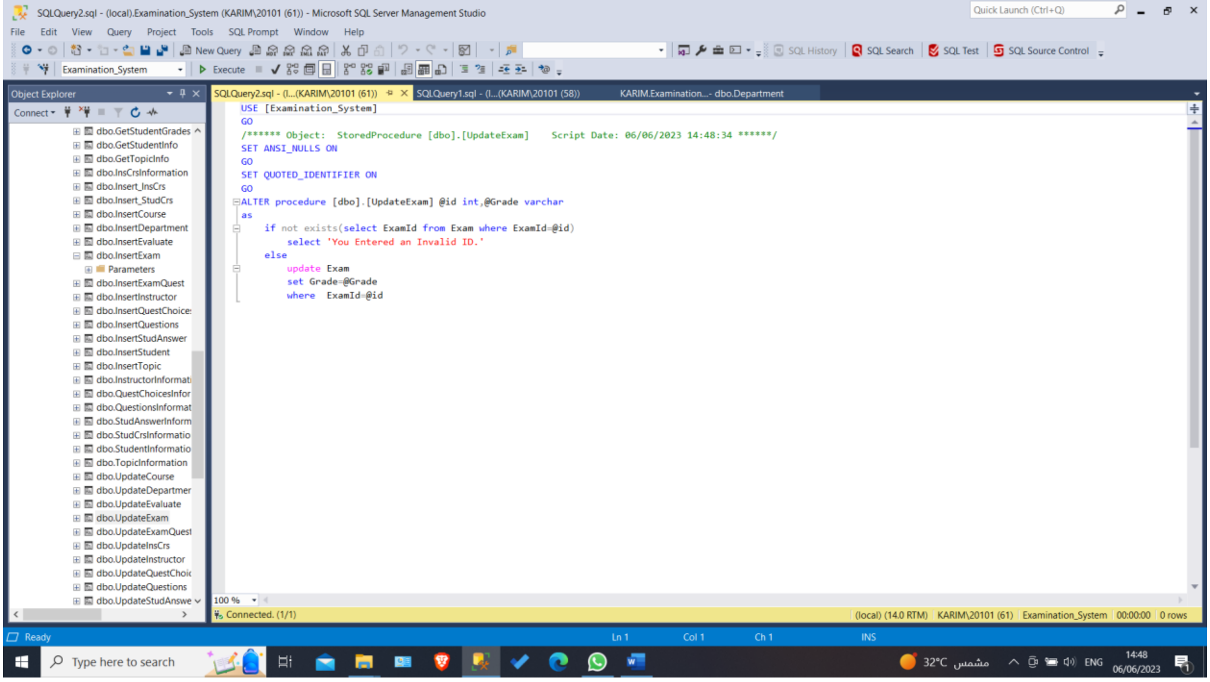
a) Insert



```
USE [Examination_System]
GO
/****** Object:  StoredProcedure [dbo].[InsertExam]    Script Date: 06/06/2023 14:48:06 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[InsertExam] @Id int,@name varchar(50),@status varchar(20),@Exdate date,
@grade varchar(20),@ExFees int,@sid int
 as
 if not exists(select ExamId from Exam where ExamId=@Id)
    if @Id < 0
        select 'Invalid ID.'
    else
        insert into Exam(ExamId,ExamName,Status,ExamDate,Grade,ExamFees,SID)
        values(@Id,@name,@status,@Exdate,@grade,@ExFees,@sid)
 else
    select 'The ID You Entered Is Already In Use.'
```

SQLQuery1.sql - (local).Examination_System (KARIM\20101 (58)) - Microsoft SQL Server Management Studio
Object Explorer
SQLQuery1.sql - (l...(KARIM\20101 (58))
KARIM.Examination...- dbo.Department
Connected. (1/1)
Ready

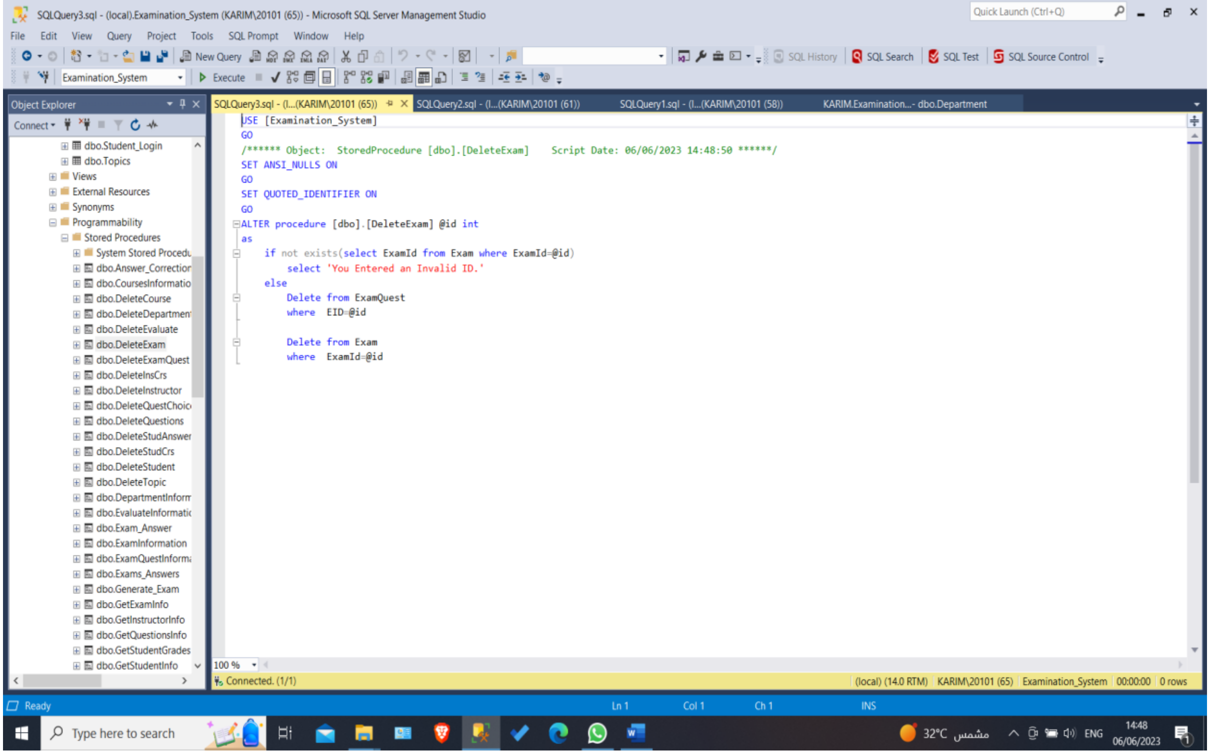
b) Update



```
USE [Examination_System]
GO
/****** Object:  StoredProcedure [dbo].[UpdateExam]    Script Date: 06/06/2023 14:48:34 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[UpdateExam] @id int,@Grade varchar
as
    if not exists(select ExamId from Exam where ExamId=@id)
        select 'You Entered an Invalid ID.'
    else
        update Exam
        set Grade=@Grade
        where  ExamId=@id
```

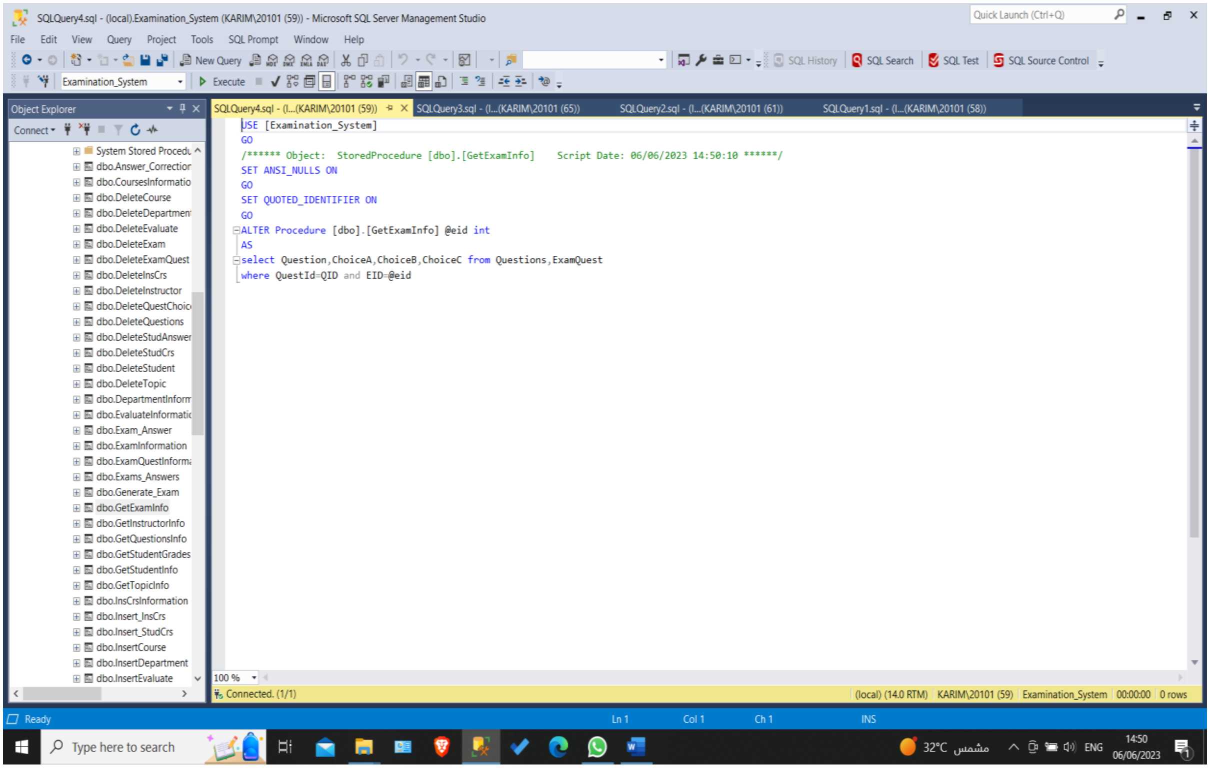
SQLQuery2.sql - (local).Examination_System (KARIM\20101 (61)) - Microsoft SQL Server Management Studio
Object Explorer
SQLQuery2.sql - (l...(KARIM\20101 (61))
SQLQuery1.sql - (l...(KARIM\20101 (58))
KARIM.Examination...- dbo.Department
Connected. (1/1)
Ready

c) Delete



```
USE [Examination_System]
GO
/****** Object:  StoredProcedure [dbo].[DeleteExam]    Script Date: 06/06/2023 14:48:50 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[DeleteExam] @id int
as
    if not exists(select ExamId from Exam where ExamId=@id)
        select 'You Entered an Invalid ID.'
    else
        Delete from ExamQuest
        where  EID=@id

        Delete from Exam
        where  ExamId=@id
```

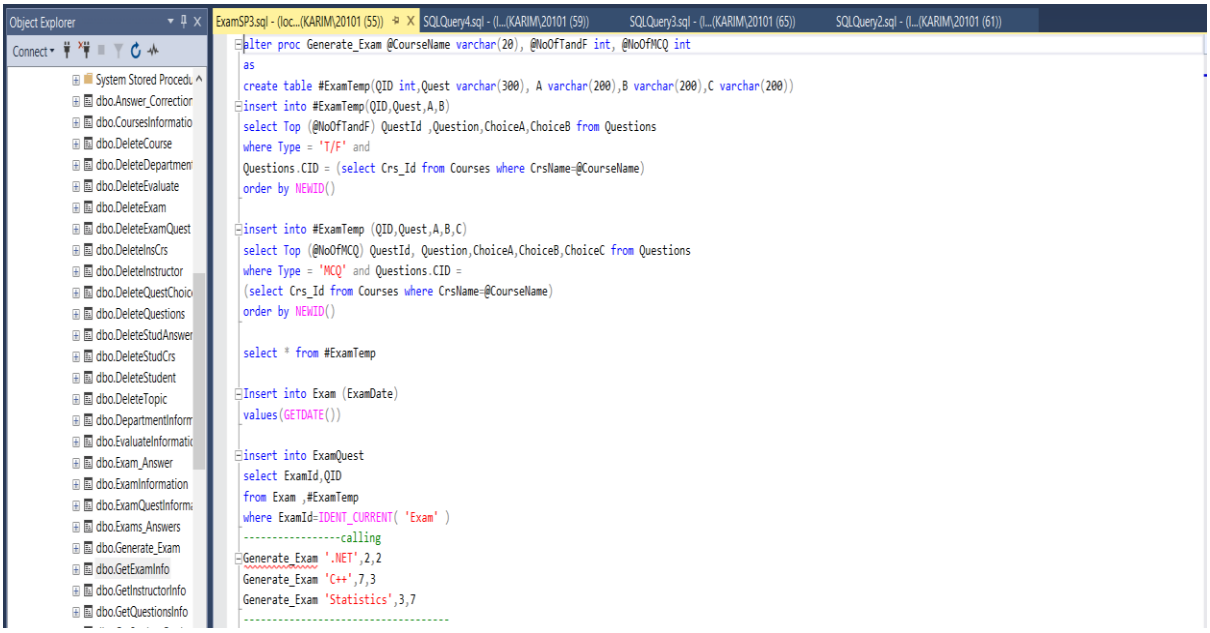
d) Select



```
USE [Examination_System]
GO
/****** Object:  StoredProcedure [dbo].[GetExamInfo]    Script Date: 06/06/2023 14:50:10 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[GetExamInfo] @eid int
AS
select Question,ChoiceA,ChoiceB,ChoiceC from Questions,ExamQuest
where QuestId=QID and EID=@eid
```

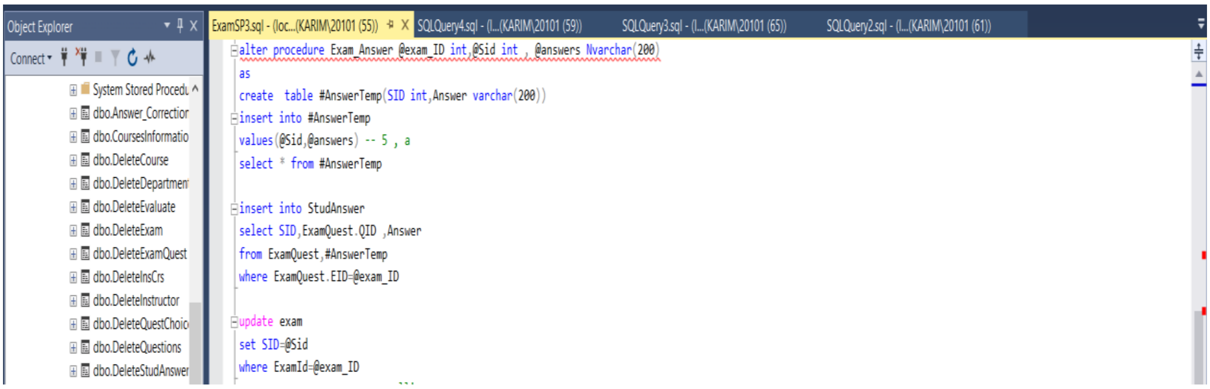
B. Create 3 core stored procedure which maintain exam generation, taking answers, and answer correction.

a. Generate Exam Procedure
Which takes course ID and number of MCQ and T/F questions as parameters and generate exam for the student.



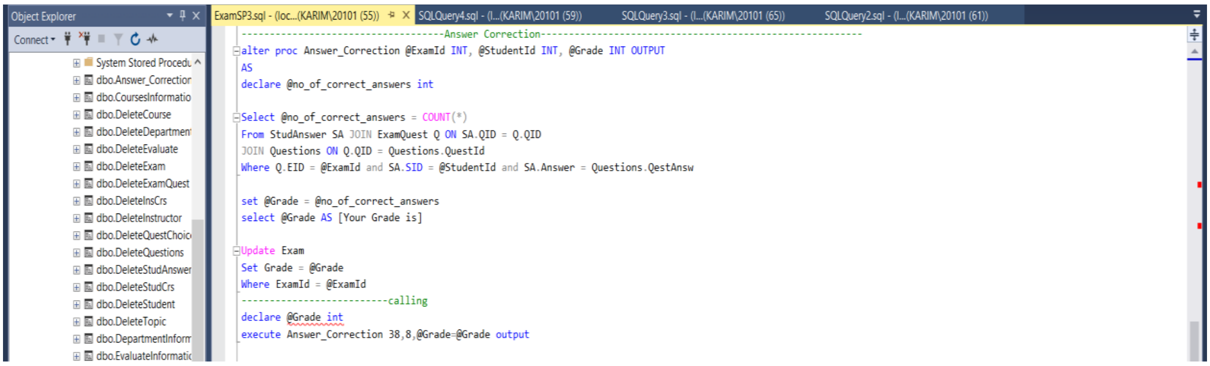
```sql
alter proc Generate_Exam @CourseName varchar(20), @NoOfTandF int, @NoOfMCQ int
as
create table #ExamTemp(QID int,Quest varchar(300), A varchar(200),B varchar(200),C varchar(200))
insert into #ExamTemp(QID,Quest,A,B)
select Top (@NoOfTandF) QuestId ,Question,ChoiceA,ChoiceB from Questions
where Type = 'T/F' and
Questions.CID = (select Crs_Id from Courses where CrsName=@CourseName)
order by NEWID()

insert into #ExamTemp (QID,Quest,A,B,C)
select Top (@NoOfMCQ) QuestId, Question,ChoiceA,ChoiceB,ChoiceC from Questions
where Type = 'MCQ' and Questions.CID =
(select Crs_Id from Courses where CrsName=@CourseName)
order by NEWID()

select * from #ExamTemp

Insert into Exam (ExamDate)
values(GETDATE())

insert into ExamQuest
select ExamId,QID
from Exam ,#ExamTemp
where ExamId=IDENT_CURRENT( 'Exam' )
----------------calling
Generate_Exam '.NET',2,2
Generate_Exam 'C++',7,3
Generate_Exam 'Statistics',3,7
-----------------------------------
```

b. Exam Answer Procedure
Which take exam id, student id and Answers as parameters then insert these values into Student answer table.



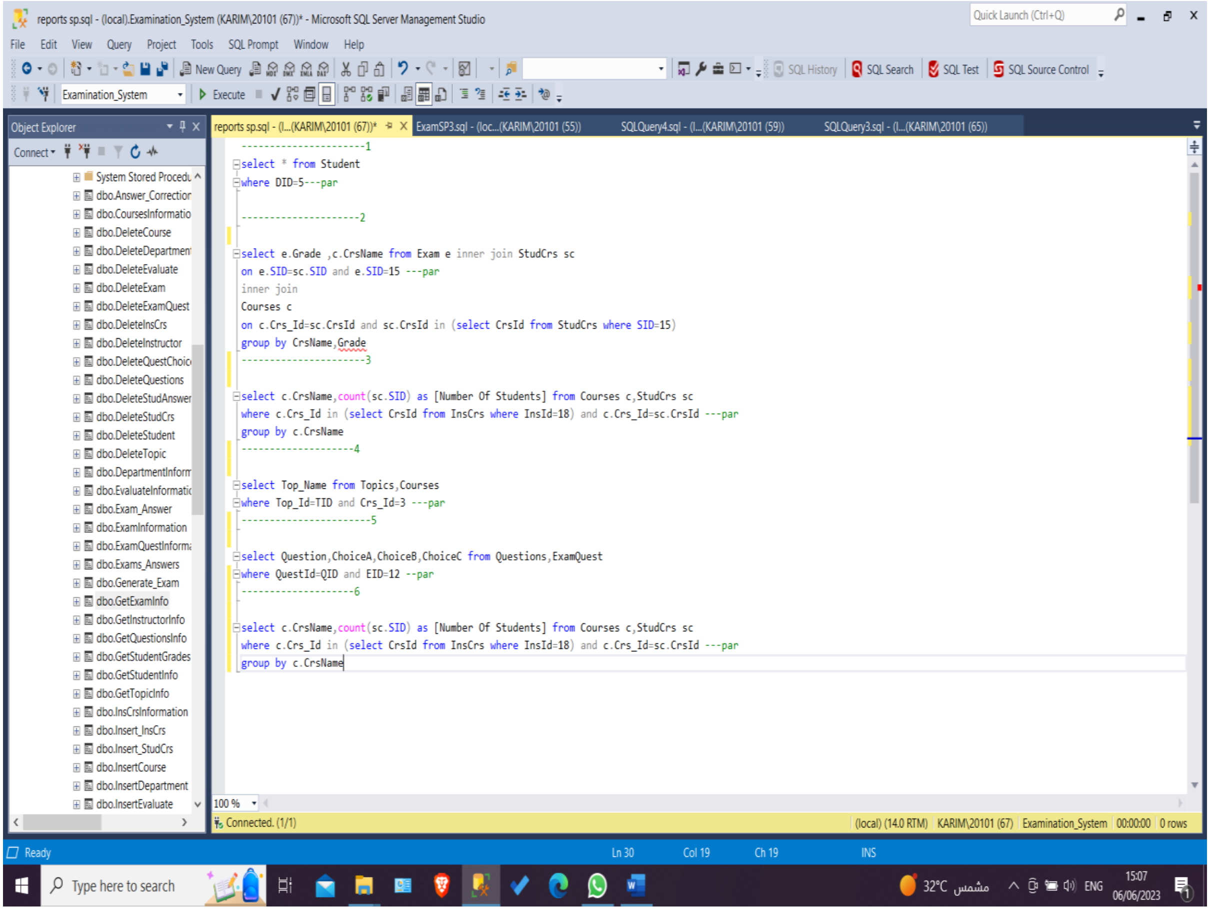
```sql
alter procedure Exam_Answer @exam_ID int,@Sid int , @answers Nvarchar(200)
as
create  table #AnswerTemp(SID int,Answer varchar(200))
insert into #AnswerTemp
values(@Sid,@answers) -- 5 , a
select * from #AnswerTemp

insert into StudAnswer
select SID,ExamQuest.QID ,Answer
from ExamQuest,#AnswerTemp
where ExamQuest.EID=@exam_ID

update exam
set SID=@Sid
where ExamId=@exam_ID
```

c. Answer Correction Procedure
Which take exam id, student id as parameters then evaluate the answers and insert grade.



```sql
-----------------------------------Answer Correction-----------------------------------------------------------
alter proc Answer_Correction @ExamId INT, @StudentId INT, @Grade INT OUTPUT
AS
declare @no_of_correct_answers int

Select @no_of_correct_answers = COUNT(*)
From StudAnswer SA JOIN ExamQuest Q ON SA.QID = Q.QID
JOIN Questions ON Q.QID = Questions.QuestId
Where Q.EID = @ExamId and SA.SID = @StudentId and SA.Answer = Questions.QestAnsw

set @Grade = @no_of_correct_answers
select @Grade AS [Your Grade is]

Update Exam
Set Grade = @Grade
Where ExamId = @ExamId
-------------------------calling
declare @Grade int
execute Answer_Correction 38,8,@Grade=@Grade output
```

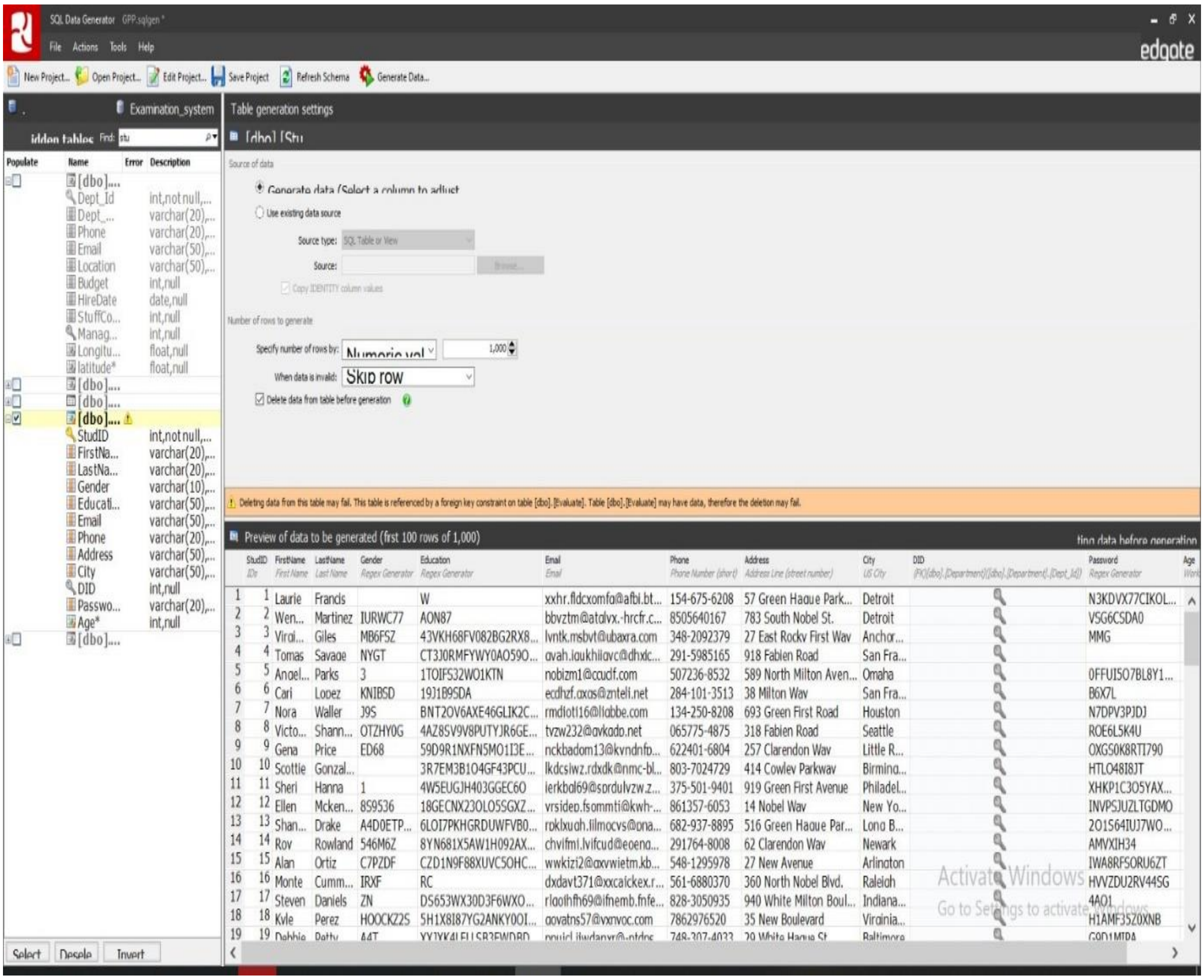
C. Creating 6 stored procedure for SSRS reports.



```sql
---------------------1
select * from Student
where DID=5---par


--------------------2

select e.Grade ,c.CrsName from Exam e inner join StudCrs sc
on e.SID=sc.SID and e.SID=15 ---par
inner join
Courses c
on c.Crs_Id=sc.CrsId and sc.CrsId in (select CrsId from StudCrs where SID=15)
group by CrsName,Grade
---------------------3

select c.CrsName,count(sc.SID) as [Number Of Students] from Courses c,StudCrs sc
where c.Crs_Id in (select CrsId from InsCrs where InsId=18) and c.Crs_Id=sc.CrsId ---par
group by c.CrsName
-------------------4

select Top_Name from Topics,Courses
where Top_Id=TID and Crs_Id=3 ---par
---------------------5

select Question,ChoiceA,ChoiceB,ChoiceC from Questions,ExamQuest
where QuestId=QID and EID=12 --par
-------------------6

select c.CrsName,count(sc.SID) as [Number Of Students] from Courses c,StudCrs sc
where c.Crs_Id in (select CrsId from InsCrs where InsId=18) and c.Crs_Id=sc.CrsId ---par
group by c.CrsName
```

- ## Red gate tools

1- Using data generator tool, we had filled our database with data.



SQL Data Generator GPP.sqlgen *
File Actions Tools Help
New Project... Open Project... Edit Project... Save Project Refresh Schema Generate Data...
edgate
Examination_system
iddon tables Find: stu
Select Desele Invert

Populate	Name	Error	Description
⊟□	[dbo]....		
	Dept_Id		int,not null,...
	Dept_...		varchar(20),...
	Phone		varchar(20),...
	Email		varchar(50),...
	Location		varchar(50),...
	Budget		int,null
	HireDate		date,null
	StuffCo...		int,null
	Manag...		int,null
	Longitu...		float,null
	latitude*		float,null
⊞□	[dbo]....		
⊞□	[dbo]....		
⊟☑	[dbo].... ⚠		
	StudID		int,not null,...
	FirstNa...		varchar(20),...
	LastNa...		varchar(20),...
	Gender		varchar(10),...
	Educati...		varchar(50),...
	Email		varchar(50),...
	Phone		varchar(20),...
	Address		varchar(50),...
	City		varchar(50),...
	DID		int,null
	Passwo...		varchar(20),...
	Age*		int,null
⊞□	[dbo]....		

Table generation settings

[dbo].[Stu

Source of data

● Generate data (Select a column to adjust

○ Use existing data source

Source type: SQL Table or View

Source:

☑ Copy IDENTITY column values

Number of rows to generate

Specify number of rows by: Numeric val 1,000

When data is invalid: Skip row

☑ Delete data from table before generation

⚠ Deleting data from this table may fail. This table is referenced by a foreign key constraint on table [dbo].[Evaluate]. Table [dbo].[Evaluate] may have data, therefore the deletion may fail.

Preview of data to be generated (first 100 rows of 1,000)

ting data before generation

	StudID IDs	FirstName First Name	LastName Last Name	Gender Regex Generator	Education Regex Generator	Email Email	Phone Phone Number (short)	Address Address Line (street number)	City US City	DID (FK)[dbo].[Department]([dbo].[Department].[Dept_Id])	Password Regex Generator	Age Work
1	1	Laurie	Francis		W	xxhr.fldcxomfa@afbi.bt...	154-675-6208	57 Green Hague Park...	Detroit		N3KDVX77CIKOL...	
2	2	Wen...	Martinez	IURWC77	AON87	bbvztm@atalvx.-hrcfr.c...	8505640167	783 South Nobel St.	Detroit		VSG6CSDA0	
3	3	Virgi...	Giles	MB6FSZ	43VKH68FV082BG2RX8...	lvntk.msbvt@ubaxra.com	348-2092379	27 East Rocky First Way	Anchor...		MMG	
4	4	Tomas	Savage	NYGT	CT3J0RMFYWY0AO59O...	avah.iaukhiiavc@dhxdc...	291-5985165	918 Fabien Road	San Fra...			
5	5	Angel...	Parks	3	1TOIFS32WO1KTN	nobizm1@ccudf.com	507236-8532	589 North Milton Aven...	Omaha		0FFUI5O7BL8Y1...	
6	6	Cari	Lopez	KNIBSD	19J1B9SDA	ecdhzf.axas@znteli.net	284-101-3513	38 Milton Way	San Fra...		B6X7L	
7	7	Nora	Waller	J9S	BNT2OV6AXE46GLIK2C...	rmdioti16@liabbe.com	134-250-8208	693 Green First Road	Houston		N7DPV3PJDJ	
8	8	Victo...	Shann...	OTZHY0G	4AZ8SV9V8PUTYJR6GE...	tvzw232@avkado.net	065775-4875	318 Fabien Road	Seattle		ROE6L5K4U	
9	9	Gena	Price	ED68	59D9R1NXFN5MO1I3E...	nckbadom13@kvndnfb...	622401-6804	257 Clarendon Way	Little R...		OXGS0K8RTI790	
10	10	Scottie	Gonzal...		3R7EM3B1O4GF43PCU...	lkdcsiwz.rdxdk@nmc-bl...	803-7024729	414 Cowley Parkway	Birming...		HTLO48I8JT	
11	11	Sheri	Hanna	1	4W5EUGJH403GGEC6O	ierkbal69@sprdulvzw.z...	375-501-9401	919 Green First Avenue	Philadel...		XHKP1C3O5YAX...	
12	12	Ellen	Mcken...	8S9536	18GECNX23OLO5SGXZ...	vrsidep.fsommti@kwh-...	861357-6053	14 Nobel Way	New Yo...		INVPSJUZLTGDMO	
13	13	Shan...	Drake	A4D0ETP...	6LOI7PKHGRDUWFVB0...	rpklxuah.iilmocvs@pna...	682-937-8895	516 Green Hague Par...	Long B...		2O1S64IUJ7WO...	
14	14	Roy	Rowland	546M6Z	8YN681X5AW1H092AX...	chvifmi.lvifcud@eoena...	291764-8008	62 Clarendon Way	Newark		AMVXIH34	
15	15	Alan	Ortiz	C7PZDF	CZD1N9F88XUVC5OHC...	wwkizi2@axvwietm.kb...	548-1295978	27 New Avenue	Arlington		IWA8RFSORU6ZT	
16	16	Monte	Cumm...	IRXF	RC	dxdavt371@xxcaickex.r...	561-6880370	360 North Nobel Blvd.	Raleigh		HVVZDU2RV44SG	
17	17	Steven	Daniels	ZN	DS653WX30D3F6WXO...	rlaolhfh69@ifnemb.fnfe...	828-3050935	940 White Milton Boul...	Indiana...		4AO1	
18	18	Kyle	Perez	HOOCKZ2S	5H1X8I87YG2ANKY0OI...	aovatns57@vxnvoc.com	7862976520	35 New Boulevard	Virginia...		H1AMF35Z0XNB	
19	19	Debbie	Patty	A4T	YXJYK4LFLLSB3FWDBD	pouicl.itwdanxr@-ntdns	748-307-4033	20 White Hague St	Baltimore		G9D1MIPA	

- **C# Language**

  1- Using C# language, we created a desktop application to maintain exam process and connect the application with our examination system database.

- **SSRS**

  1- We created 6 Crystal Reports using SSRS.

- **Power BI & Tableau**

  1- Using data visualization tools, we created more than 20 dashboards to provide valuable insights into the performance of an examination system and highlight areas where improvements can be made to support student success.

Conclusion

Our examination system project using SQL, SSRS, Power BI, and Tableau has been a successful demonstration of our skills as Power BI developers. By leveraging these tools, we were able to create a comprehensive system for managing examination data, generating reports, and providing valuable insights into student performance.

By using SQL, we were able to design and implement a robust database schema that can handle large volumes of data and provide efficient data retrieval. SSRS allowed us to create custom reports that can be tailored to the specific needs of different stakeholders, such as administrators or teachers.

Power BI and Tableau provided powerful data visualization capabilities, allowing us to create interactive dashboards that provide real-time insights into student performance and help decision-makers make data-driven decisions. By leveraging these tools, we were able to create a user-friendly system that is accessible to a broad range of users, from administrators to teachers to students.

Overall, our examination system project demonstrates the power of using SQL, SSRS, Power BI, and Tableau in combination to create a comprehensive system for managing examination data and generating valuable insights. We hope that this documentation file serves as a useful resource for those interested in using these tools to develop similar projects, and we look forward to applying our skills to future data analysis projects.